Data Structures and Algorithms — Lab 1

Objective

- Basic concepts of 2-D array
- Revision of classes
- Introduction to *Attributes* and *Methods*

2-D Array

You've worked one-dimensional array in Programming Fundamental courses. An array keeps track of multiple pieces of similar information in linear order. However, data associated with some applications like images, statistical survey are represented in two dimensions – we use 2-D arrays in such applications. A two-dimensional array is nothing more than the *array of arrays.*

Matrix

Data in some applications can only be represented in 2-D arrays. In elementary classes school, you've studied *Matrices* (singular: Matrix). A matrix is a rectangular array of mathematical objects such as whole numbers, real numbers and complex numbers etc. A Matrix can be represented using a 2-D array.

Lab Task 1

Mr. Ghuman teaches Mathematics courses to grade 8 in Punjab School. He wants to provide simple console-based application to his students. This application can calculate addition, subtraction, constant multiplication, constant addition, constant subtraction, multiplication etc. He asks you to give him a favor to accomplish this task.

The application/program should be generic, and ask the student to give the number of rows and columns of two matrices. It'll create two matrices and then it will ask the operation to perform.

Students are required to perform the lab task during lab timings. Create *C++ class* for attributes/methods and check their functionality in `main()` by creating a *menu*. Please note that attributes of a class are the variables and methods of a class are the functions written in it. Make your class according to the template given below:

```
struct TwoDArray
{
private:
      //write your attributes (varibales) here
public:
      //write your methods (functions) here
};
```

Options to be displayed as menu in your main() program:

```
Options
1: Input Matrix
2: Display Matrix
```

```
3: Add Matrices
4: Subtract Matrices
5: Add a Constant value to Matrix
6: Multiply Matrices
7: Exit
```

Note: The program should be menu driven and only exit upon entering 7.

Lab Task 2

Repeat task 1 – this time all inputs must be taken from the file. Students are required to perform the lab task during lab timings